

Задача funcsp_rotateC-Поворот

Задача funcsp_rotateC: Поворот

Прямоугольник со сторонами, параллельными осям, задан координатами двух точек - левой верхней и правой нижней.

Написать и использовать функцию **void rotateC(int * x1, int * y1, int * x2, int * y2)**, которая поворачивает прямоугольник вокруг его центра (пересечения диагоналей).

Гарантируется, что тесты будут оставлять прямоугольник после поворота в целых координатах. Ничего округлять не нужно.

Формат входных данных

4 целых числа через пробел - x и y координаты левой верхней и правой нижней точки исходного прямоугольника.

Формат результата

4 целых числа через пробел - x и y координаты левой верхней и правой нижней точки после поворота.

Примеры

Входные данные

```
0 6 10 4
```

Результат работы

```
4 10 6 0
```

Задача f_temper-f_temper

Полный балл:	30
Штраф за послылку:	1
Ограничение времени:	4 с
Ограничение реального времени:	10 с
Ограничение памяти:	64М

Температура

Написать функцию , которая переводит температуру по Фаренгейту, в температуру по Цельсию. Измерения - в целых числах. Формула для перевода: Градусы по Цельсию = (градусы по Фаренгейту - 32) / 1.8

```
float temFC(int faren);
```

,

Для отладки использовать функцию main() и написанную функцию float temFC(int faren).

Для проверки отправлять только функцию!!!

Задача f_printTime-f_printTime

Местное время

Московское время **hMos** часов, разница по времени с городом N **h** часов. Если **h < 0**, город N находится восточнее Москвы, если наоборот - западнее.

Самолет вылетает из Москвы в **hMos** часов 5 числа некоторого месяца и летит **hplane** часов.

Необходимо узнать во сколько прилетает самолет в город N по местному времени.

Написать функцию

```
void printLocalTime(int hMos, int h, int hplane);
```

,

которая получает: Московское время (hMos), разницу во времени с городом N(h), время полета(hplane) и печатает местное время и дату прилета самолета (только число месяца) в город N.

В Москве и в городе N 24-часовой отсчет времени.

Для отладки использовать функцию main() и написанную функцию printLocalTime().

Для проверки отправлять только функцию!!! Не использовать конструкцию **if**, циклы, массивы, рекурсию....!!!

Примеры

Входные данные

```
17 7 8
```

Результат работы

```
18 5
```

Входные данные

```
9 -7 10
```

Результат работы

```
2 6
```

Задача f_middle-Новый центр отрезка

Задача f_middle: Новый центр отрезка

Отрезок на прямой OX задан своими координатами (int). Написать и использовать функцию **void toMiddle(int * x1, int * x2, int middle)**, которая перемещает отрезок так, чтобы его центр оказался в точке middle.

Округление новых координат производить по математическим правилам округления.

Посылать нужно только функцию.

Формат входных данных

3 числа через пробел - координаты начала, конца исходного отрезка и точки middle.

Формат результата

2 числа через пробел - координаты начала и конца отрезка после перемещения.

Примеры

Входные данные

```
0 2 2
```

Результат работы

```
1 3
```

Входные данные

```
-1 8 0
```

Результат работы

```
-4 5
```

Задача func_m_geron-Формула Герона

Функция

Можно вычислить площадь треугольника **s** по трем его сторонам.

Напишите и проверьте функции:

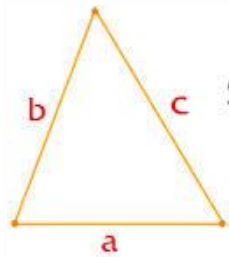
```
float length(int x1, int y1, int x2, int y2);
```

, которая вычисляет расстояние между 2 точками.
Длина не может быть отрицательной.

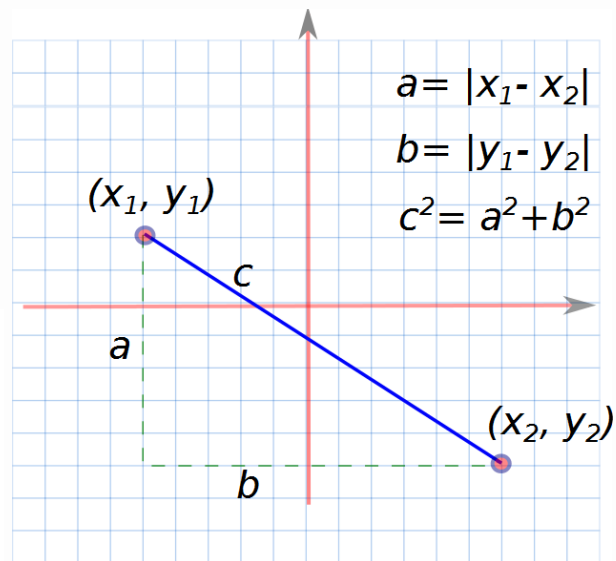
$$c^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

```
float s3(int x1, int y1, int x2, int y2, int x3, int y3);
```

, которая вычисляет площадь треугольника по 3 точкам по формуле



$$S = \sqrt{p(p-a)(p-b)(p-c)}$$
$$p = \frac{a+b+c}{2}$$



Входные данные

Координаты 3 точек x1 y1 x2 y2 x3 y3

Выходные данные

Площадь треугольника

Примеры

Вход	Выход
3 0 0 4 0 0	6

Задача f_dsum1-Сумма цифр(1)

Сумма четных и нечетных цифр

Дано восьмизначное десятичное число.

Написать функцию

```
void printOddEvent(unsigned int number);
```

,

которая делит это число на 2 и печатает сумму цифр, стоящих на четных местах этого числа, а через пробел - сумму цифр, стоящих на нечетных местах этого числа. Цифра с номером 1 - самый младший разряд числа.

Для отладки использовать функцию main().

Для проверки отправлять только функцию!!!

Примеры

Входные данные

24242424

Результат работы

8 4

Задача f_shifr2-f_shifr2

Шифр (2)

Слово из четырех заглавных латинских букв необходимо зашифровать 8-значным целым числом. Каждая латинская буква кодируется двумя десятичными цифрами этого числа.

Код букв соответствует положению буквы в латинском алфавите.

Написать функцию

```
unsigned int codeToNumber(char l1, char l2, char l3, char l4);
```

, которая кодирует слово из 4-х заглавных латинских букв в 8-разрядное число.

Функция будет использоваться в программе:

```
#include <stdio.h>
unsigned int codeToNumber(char l1, char l2, char l3, char l4);
int main(){
    char l1, l2, l3 ,l4;
    unsigned long long number;
    scanf("%c%c%c%c", &l1, &l2, &l3, &l4 );

    printf("%08u\n", codeToNumber(l1, l2, l3, l4));
}
```

Конструкции **if**, циклы, массивы, рекурсию не использовать!!! После написания и проверки программы с функцией, проверяющей систему отправлять только функцию!!!

Выходные данные. Одно целое восьмизначное число - закодированное слово.

Примеры

Входные данные

AAAA

Результат работы

01010101

Задачи f_shifr1-f_shifr1

Шифр

Слово из четырех заглавных латинских букв зашфровано 8-значным целым числом. Каждая латинская буква кодируется двумя десятичными цифрами этого числа.

Код букв соответствует положению буквы в латинском алфавите.

Написать функцию

```
void deCode(unsigned int kodeNumber, char *l1, char *l2, char *l3, char *l4);
```

, где **kodeNumber** - код слова в виде числа, ***l1**, ***l2**, ***l3**, ***l4** - адреса переменных для записи соответствующих букв.

Задача f_tret-f_tret

Третъ

Отрезок на прямой задан своими координатами (float).

Написать функцию

```
void oneTr(float* x1, float* x2);
```

,

, которая "обрезает" отрезок сначала на $1/3$ и с конца на $1/3$. При этом $x1$ и $x2$ получают новые координаты.

Например, для точек 0 и 12 новые координаты будут 4 и 8, и для 12 и 0 новые координаты будут 4 и 8.

Для отладки использовать функцию `main()` и написанную функцию `float temFC(int faren)`.

Для проверки отправлять только функцию!!!

Конструкцию `if`, циклы, массивы, рекурсию не использовать!!!

Задача funcsp_mirror1_с-Отобразить точку

Задача funcsp_mirror1_с: Отобразить точку

Написать и использовать функцию **void mirror(int * x1, int * y1)**, которая отображает точку с координатами x,y относительно оси Y.

Формат входных данных

2 целых числа через пробел - x и y координаты точки до отображения.

Формат результата

2 целых числа через пробел - x и y координаты точки после отображения.

Примеры

Входные данные

```
3 2
```

Результат работы

```
-3 2
```

Задача funcpr_mirror2_с-Отобразить прямоугольник

Задача funcpr_mirror2_с: Отобразить прямоугольник

Написать и использовать функцию **void mirror(int * x1, int * y1, int * x2, int * y2)**, которая отображает отрезок относительно оси X.

Формат входных данных

4 целых числа через пробел - x и y координаты концов отрезка до отображения.

Формат результата

4 целых числа через пробел - x и y координаты концов отрезка после отображения.

Примеры

Входные данные

```
3 2 7 -1
```

Результат работы

```
3 -2 7 1
```

Задача funcpr_rotateC-Поворот

Задача funcpr_rotateC: Поворот

Прямоугольник со сторонами, параллельными осям, задан координатами двух точек - левой верхней и правой нижней.

Написать и использовать функцию **void rotateC(int * x1, int * y1, int * x2, int * y2)**, которая поворачивает прямоугольник вокруг его центра (пересечения диагоналей).

Гарантируется, что тесты будут оставлять прямоугольник после поворота в целых координатах. Ничего округлять не нужно.

Формат входных данных

4 целых числа через пробел - x и y координаты левой верхней и правой нижней точки исходного прямоугольника.

Формат результата

4 целых числа через пробел - x и y координаты левой верхней и правой нижней точки после поворота.

Примеры

Входные данные

```
0 6 10 4
```

Результат работы

```
4 10 6 0
```